**Formación Desarrollo Web**

1. Cómo funcionan los sitios web

Desde el momento en que el usuario abre un navegador web, introduce una URL y presiona ENTER, se cumple el siguiente proceso:

1. La URL se “resuelve”
2. Una petición es enviada al servidor del sitio web
3. La respuesta del servidor es analizada
4. La página se renderiza y se muestra en el navegador

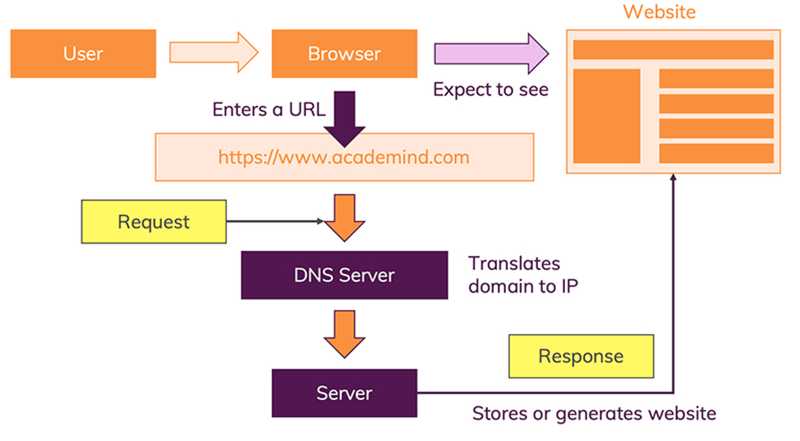


Ilustración 1 - Cómo funcionan los sitios web

1. *Server-side vs Browser-side*

Diferencia entre las dos “partes” cuando se habla de la web:

* 1. *Server-side*: lenguajes de programación del lado del servidor que no se ejecutan en navegadores, pero se pueden ejecutar en un ordenador normal (un servidor es, a fin de cuentas, un ordenador normal). Algunos ejemplos de lenguajes de programación del lado del servidor son Node.js, PHP y Python
  2. *Browser-side*: lenguajes que se ejecutan en el navegador (HTML, CSS y JavaScript)

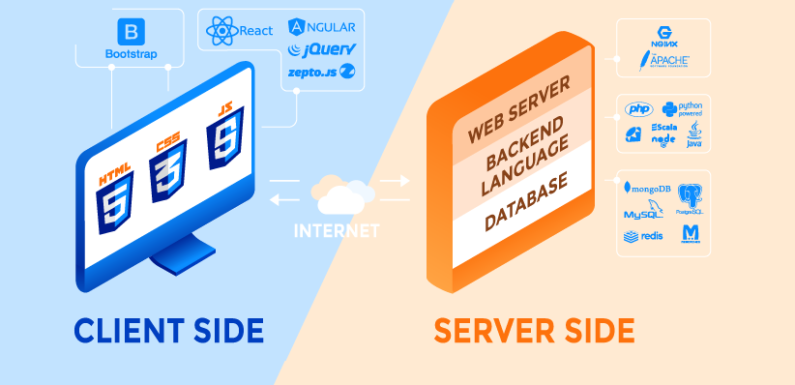


Ilustración 2 - Server-side vs Browser-side

1. Tecnologías del lado del cliente
   1. HTML: lenguaje de marcado (*HyperText Markup Language*) estándar usado para construir páginas web
   2. CSS: lenguaje de hojas de estilos en cascada (*Cascading Style Sheets*). Describe cómo serán mostrados los elementos HTML
   3. JavaScript: lenguaje de programación que permite realizar actividades complejas es una página web
2. HTML
   * Editores
   * Elementos
   * Atributos
   * Encabezados
   * Párrafos
   * Formato de texto
   * Comentarios
   * CSS (abreboca)
   * *Links*
   * Imágenes
   * Tablas
   * Listas
   * Elementos *block* e *inline*
   * Clases e identificadores únicos
   * Iframes
   * Rutas de archivo
   * Etiqueta *head*
   * Entidades
   * Guía de estilos
3. CSS
   * Sintaxis
   * Selectores
   * Cómo incluir CSS
   * Comentarios
   * Colores
   * *Backgrounds*
   * Bordes
   * Márgenes
   * *Paddings*
   * Alto y ancho
   * *Box model*
   * Texto
   * Fuentes
   * Iconos
   * *Links*
   * Listas
   * Tablas
   * *Display*
   * *Max-width*
   * Posición
   * *Overflow*
   * *Float*
   * *Inline-block*
   * Alinear
   * Combinadores
   * Pseudo clases
   * Pseudo elementos
   * Opacidad y transparencia
   * Selectores de atributos
   * Unidades
4. JavaScript
   * Introducción
   * *Output*
   * *Statementents*
   * Sintaxis
   * Comentarios
   * Variables
   * Operadores
   * Aritmética
   * Asignación
   * Tipos de datos
   * Funciones
   * Objetos
   * Eventos
   * Cadenas
   * *Arrays*
   * Comparaciones
   * Condicionales
   * *Switch*
   * Conversiones de tipo
   * Errores
   * *Scope*
   * Palabra clave *this*
   * *Let*
   * *Const*
   * Función flecha
   * Clases
   * *Debugging*
   * Guía de estilos
   * Mejores prácticas
   * Errores comunes
   * Rendimiento
   * *Callbacks*
   * AJAX
   * Promesas
5. Tipos de aplicaciones web
   1. Estática

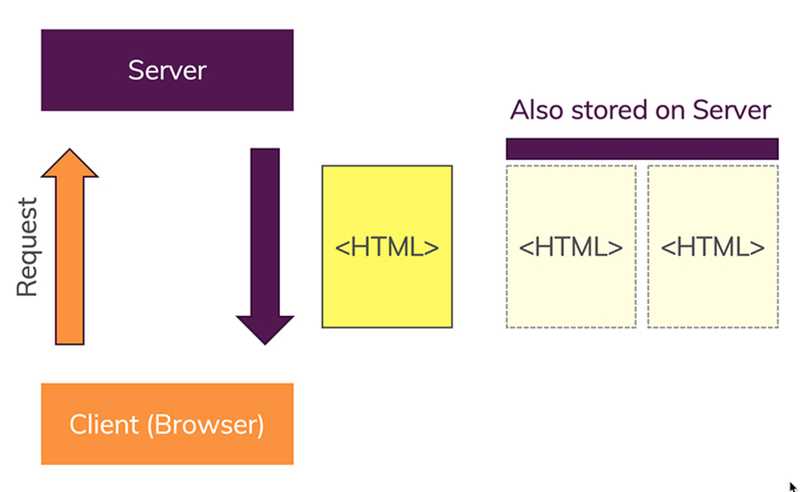


Ilustración 3 – Aplicación web estática

* 1. Dinámica

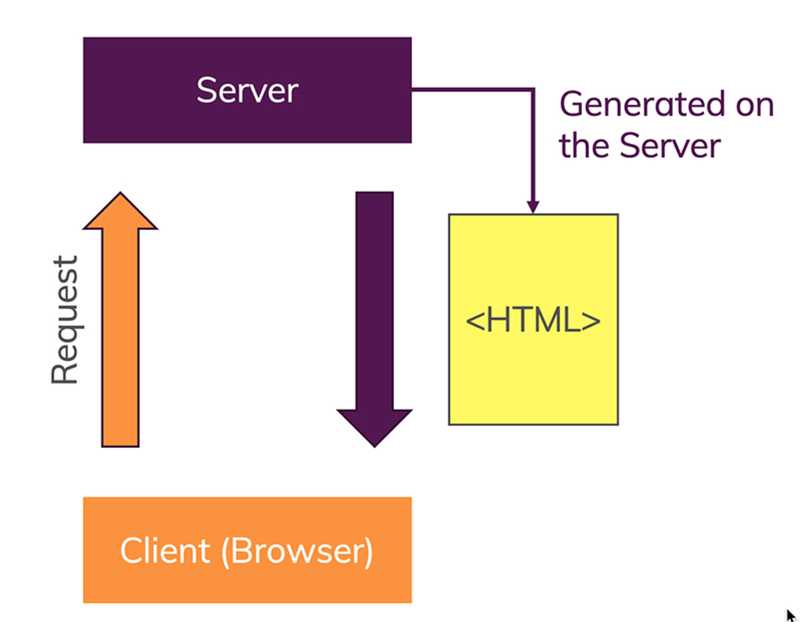


Ilustración 4 - Aplicación web dinámica

* 1. *Single Page Application* (SPA)

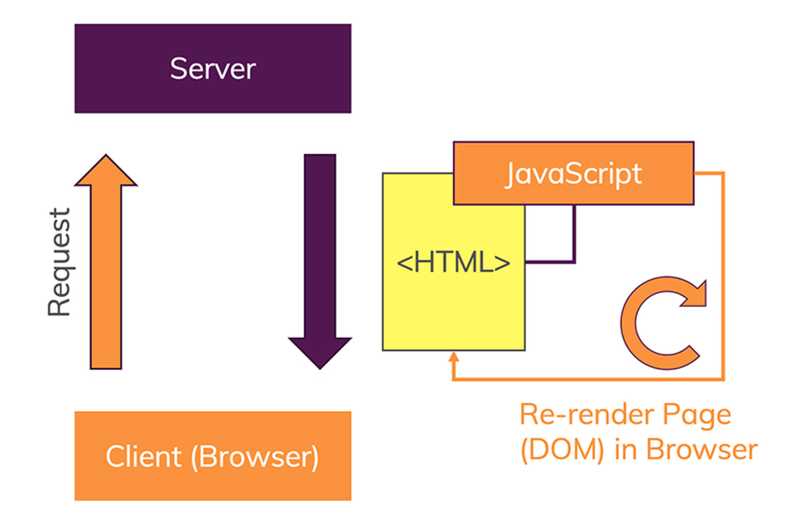


Ilustración 5 - Single Page Application (SPA)

1. Angular (*NPM* y *Visual Studio Code*)
   1. Conceptos
      * Conceptos básicos
      * Módulos
      * Componentes
      * Servicios
   2. Componentes y *Templates*
      * *Displaying data*
      * Sintaxis de *Templates*
      * *User Input*
      * Ciclo de vida de los *Hooks*
      * Interacción entre componentes
      * Estilos de componentes
      * Componentes dinámicos
      * Directivas
      * *Pipes*
   3. Formularios
      * Introducción
      * Formularios reactivos
      * Validación
      * Formularios dinámicos
   4. Observables & RxJS
      * Observables
      * La librería RxJS
      * Observables en Angular
      * Uso práctico
      * Comparación con otras técnicas
   5. *NgModules*
      * Introducción
      * Módulos usados frecuentemente
      * Componentes de entrada
      * *Providers*
      * Servicios *Singleton*
      * Módulos *Lazy loading*
      * Compartiendo módulos
      * *ngModules API*
      * *Routing & Navegation*
2. Git/GitHub/SourceTree
   1. Instalación y configuración
   2. Obtener y crear proyectos
      * *Init*
      * *Clone*
   3. *Basic Snapshotting*
      * *Add*
      * *Status*
      * *Diff*
      * *Commit*
      * *Notes*
      * *Restore*
      * *Reset*
   4. *Branching and Merging*
      * *Branch*
      * *Checkout*
      * *Switch*
      * *Merge*
      * *Stash*
   5. Compartiendo y cargando proyectos
      * *Fetch*
      * *Pull*
      * *Push*
      * *Remote*

|  |  |  |
| --- | --- | --- |
| **Tópico** | **Estimación en horas** | |
| **Preparación** | **Impartición** |
| Cómo funcionan los sitios web  *Server-side vs Browser-side*  Tecnologías del lado del cliente | 4 | 1 |
| HTML | 8 | 3 |
| CSS | 8 | 3 |
| JavaScript | 8 | 5 |
| Tipos de aplicaciones web | 2 | 1 |
| Angular (*NPM y Visual Studio Code*) | 24 | 8 |
| Git/GitHub/SourceTree | 8 | 3 |
| **Total** | **62** | **24** |